Week 5 Exercises

October 23, 2020

0.1 Exercises

Exercise 1

The following data contains the lattitude and longitude of capital cities in Canada. As you can see there are no decimal points for lat lon values which means that the locations are less accurate (check your the lat lon in your own datasets and see how many digits you have after the dot!).

Now, go back to the begining of this notebook and use this data to do the following: 1. Plot a map of Canada 2. Add cities 3. Add city names

Take a screenshot of your code and your **final** plot (you don't need to show each step as we did before, just show us the last plot which has the location of the cities and names) and upload one screenshot of both code and figure on Crowdmark.

[2]:
```
#! pip install qeds fiona geopandas xgboost gensim folium pyLDAvis descartes
```

[11]:
```
import geopandas as gpd
import matplotlib.pyplot as plt
import pandas as pd

from shapely.geometry import Point

%matplotlib inline
# activate plot theme
import qeds
qeds.themes.mpl_style();
```

[12]:
```
df_c = pd.DataFrame({
'Province':['British Columbia', 'Yukon','Alberta', 'Northwest Territories',␣
 ↪'Saskatchewan', 'Manitoba','Nunavut','Ontario',
'Québec','New Brunswick','Nova Scotia','Prince Edward Island','Newfoundland and␣
 ↪Labrador'],
'City' : ['Victoria', 'Whitehorse', 'Edmonton' ,'Yellowknife' ,'Regina',␣
 ↪'Winnipeg', 'Iqaluit', 'Toronto' ,'Québec'
          ,'Fredericton', 'Halifax' ,'Charlottetown' ,'St. John's'],
'Latitude': [48, 61, 54, 62, 50,50, 64, 44, 47, 46, 45, 53, 48],
'Longitude': [-123, -135, -113, -114,-105, -97, -69,-79,-71, -67,-64,-56, -53]
})
#notice I called this data df_c for Canada.
```

[13]:
```
#Create a new column of tuples in order to create the map
#using the longitude and latitude

df_c["Coordinates"] = list(zip(df_c.Longitude, df_c.Latitude))
#df_c.head()
```

[14]:
```
#Turn the tuple into a Shapely point object
df_c["Coordinates"] = df_c["Coordinates"].apply(Point)
#df_c.head()
```

[15]:
```
#Convert the DataFrame into a GeoDataFrame by calling the geopandas.DataFrame␣
 ↪method
gdf_c = gpd.GeoDataFrame(df_c, geometry = "Coordinates")
#gdf_c.head()
```

[17]:
```
world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres") )
#world.columns
```

[20]:
```
fig, gax = plt.subplots(figsize= (25,50) )
world.query( "iso_a3 == 'CAN' " ).plot(ax = gax, edgecolor = 'black', color =␣
 ↪'white' )
gdf_c.plot(ax=gax, color='red', alpha = 0.5)

gax.set_xlabel('longitude')
gax.set_ylabel('latitude')
gax.set_title("Canada")

gax.spines['top'].set_visible(False)
gax.spines['right'] . set_visible(False)
plt.axis('off')

for x, y, label in zip(gdf_c['Coordinates'].x, gdf_c['Coordinates'].y,␣
 ↪gdf_c['City']):
    gax.annotate(label, xy=(x,y), xytext=(4,4), textcoords='offset points')

plt.show()
```


Canada
Iqaluit
Yellowknife
Whitehorse
Edmonton
Charlottetown
Regina
Winnipeg
Victoria
St. John's
Quebec
Fredericton
Halifax
Toronto